Software Reuse and Reference Architecture Processes Study

Team: Gail McConaughy, GSFC
Mark Nestler, GST
David Isaac, Business Performance Systems/GST
Nadine Alameh, GST
Allan Doyle, Intelligent Interfaces Inc/GST

Plenary Briefing

SEEDS Second Public Workshop

June 17-20, 2002

Agenda

SEEDS Formulation

- **Recap of work to date**
  - Study Background
    - Motivation
    - Study Approach
    - Definitions (in Appendix)
  - Completed activities
    - Pre-work
    - Options and Evaluation Criteria
    - SEEDS First Public Workshop
- **Results to date**
  - Aggregate Community Opinion about Reuse
  - Aggregate Community Opinion about Reference Architecture
  - Cost Sensitivity Analysis
- **Summary**
- **Workshop plans**

Motivation of the Study

The Problem

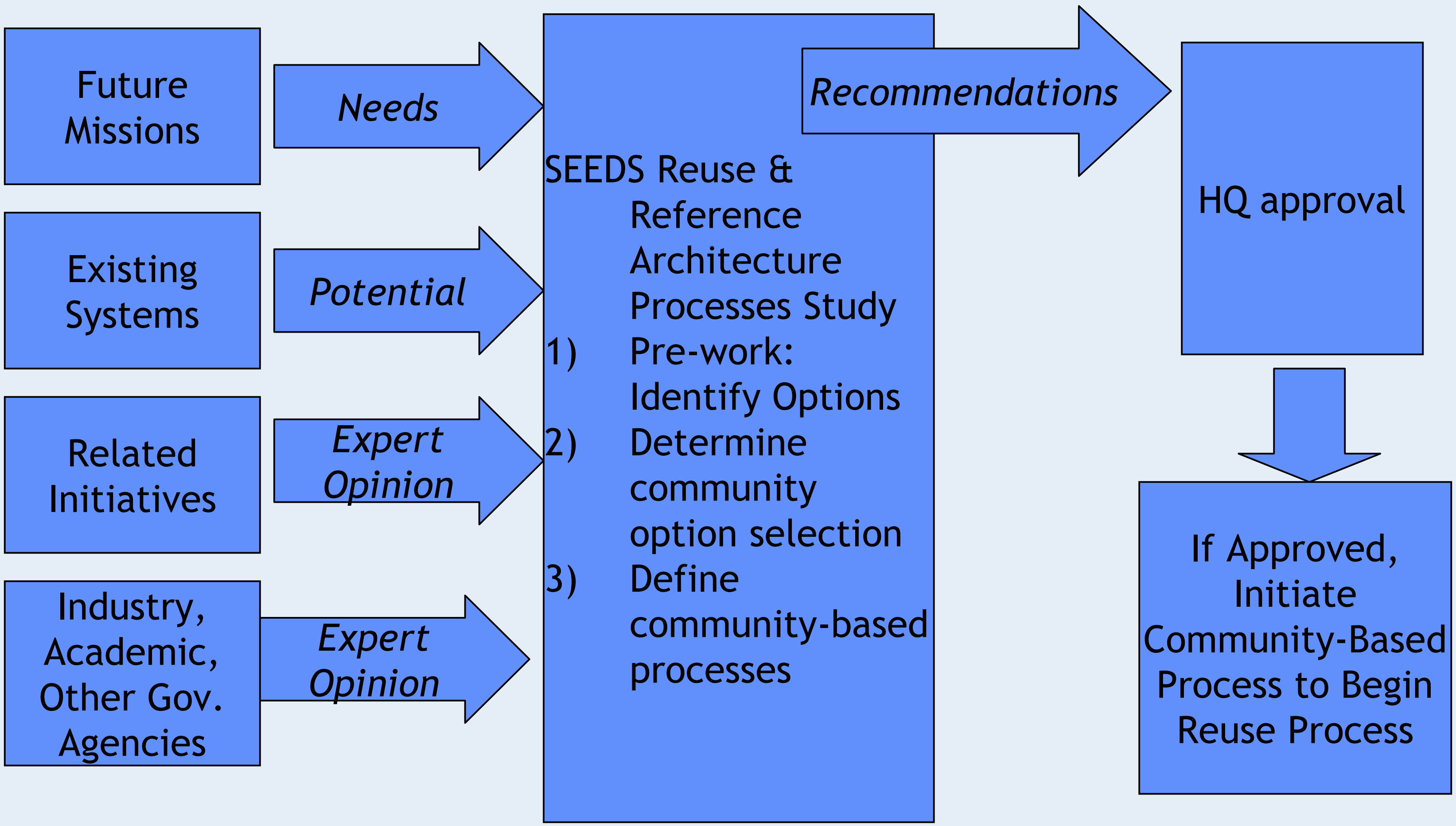
- **Need a more cost effective DISS development approach for future missions**
  - Legacy systems may well consume most of the projected ESE information systems budget
  - "Expertise" & "smallness" large positive factor in cost effective development – leverage required
- **Need a more flexible/responsive development approach**
  - Very large development efforts require rigid requirements control
  - "Smaller" efforts respond more quickly
- **Need increased and effective/accountable community participation**
  - Centralized systems do not effectively leverage community expertise
  - Community systems may not effectively leverage each other or meet critical mission requirements (e.g., long-term data retention)

The Opportunity

- **Reuse and reference architectures can reduce system development costs**
  - Reuse can leverage large base of existing ESE software, system assets and expertise
  - Reused artifacts and components require less development and testing
  - Reference architectures can enable an efficient market of components and services
- **Reuse and reference architectures can improve flexibility & responsiveness**
  - Smaller development efforts can be effectively coordinated & integrated through the ref. Architecture
  - Assembly of new systems from reused or commodity components shortens schedules
- **Reference architectures can increase community participation**
  - Enables development to be performed wherever expert resources are available
  - Ensures interoperability of independently developed components & systems
  - Provides a clear demarcation for delivered functionality

Study Approach

- Reliance on stakeholder view of supply and demand - emphasis on practical experience of actual mission to mission reuse
- Key related initiatives examined for recommendations - e.g. Carnegie Mellon SEI, OGC, OMG, ETC.
- Feedback incorporated from ESE scientific community through interviews & quarterly workshops



Completed Activities: Pre-work

- Structure Analysis & Trades
  - Initial interviews, review of documented case studies, published articles & Internet material to date
    - **Federation** NewDISS working group
    - **Related NASA initiatives**: Digital Earth Reference Model, Earth Science Modeling Framework, and the Information Power Grid, Renaissance, Open Archives Information System
    - **Current ESE systems**: ECS, TSDIS, SeaWiFS, ESIPS (Cornillon, …), DAACs (JPL, GSFC, ..), OMI, CEOS, GCMD, DIAL
    - **Future mission science systems**: Global Precipitation Mission, Total Column Ozone
    - **Related consortia:** OGC, FGDC, OMG, ISO,and CCSDS
    - **Software engineering groups**: CMU Software Engineering Institute, GSFC Software Engineering Laboratory
    - **Architecture framework initiatives**: Federal Enterprise Architecture Framework, C4ISR Architecture Framework, and the Zachman Framework, Weapons Systems Technical Architecture Working Group
    - **Government organizations** facing similar challenges: NIMA, NRO
    - **Industry Efforts:** McDonald Detweiler, NEC, GTE, Toshiba, DEC, HP, Raytheon, Fujitsu, Motorola
  - Results
    - Identification of applicable range of Reuse and Reference Architecture options
    - Identification of evaluation criteria

Range of Reuse Options

- Range of options identified **from community survey (e.g. mission system developers, CMU SEI, TSDIS/SeaWifs successes, Trends in Industry)**
- Reuse options
  - Status Quo
    - Continue employing current mix of practices including *ad hoc* "clone and own" and use of single centralized contractor
  - Improved "Clone & Own"
    - Extend current practices to enable developers to methodically copy existing assets (software & documents) and modify them as needed for use in a new system
  - Open Source Software Development
    - Selected components/systems are collaboratively developed and updated by developers across missions
  - Encapsulated Services
    - Wrap existing systems or components with network-accessible interfaces, allowing access/use by others
  - Product Lines
    - Identify, create, maintain, and evolve common core assets that can be easily integrated to build sets ("lines") of related new systems ("products")

Range of Reference Architecture Options

- Specificity
  - Status Quo
    - Continue involvement in related activities at current levels
  - Notional
    - Defines subsystems/components and allocates requirements/functionality to each
    - Examples: OpenGIS Abstract Specification Topic 12: OpenGIS Service Architecture; Reference Model for an Open Archive Information System; USIGS Objective System Architecture, OSI Reference Model
  - Concrete
    - Identifies the services (including key parameters) of each subsystem/component in lay terms
    - Examples: OpenGIS Abstract Specification Topic 13: Catalog Services; USIGS Operational Architecture; TCP/IP Tutorial (RFC 1180)
  - Specific
    - Defines the services (including all parameters) of each component in precise enough terms to build interfaces; defines the service invocation mechanism (call, post, get, etc.)
    - Examples: OpenGIS Web Map Server Implementation Specification; USIGS Technical Architecture; TCP/IP standards suite (several dozen RFCs)
- Granularity
  - Coarse: Defines external interfaces to major subsystems only
  - Medium: Defines key internal interfaces within major subsystems
  - Fine: Defines internal interfaces within applications or functional components

Evaluation Criteria

- **Potential for Increasing System Cost Savings**
  - Decreasing time-to-market
  - Improving development efficiency and productivity
  - Impact on system maintenance requirements
- **Potential for Increasing Flexibility and Responsiveness of Systems**
  - Ability to respond to new requirements
  - Ability to support new science applications
  - Ability to exploit new technologies
- **Potential for Increasing Effective and Accountable Community Participation**
  - Ability to increase community participation
  - Ability to facilitate community accountability
- **Suitability for Flight Mission Needs**
  - Fit with flight mission culture (cost & schedule emphasis)
  - Alignment of organizational requirements with current organizational structure
- **Suitability for ESIP (and similar) Needs**
  - Fit with ESIPs culture (innovation)
  - Alignment of organizational requirements with current organizational structure
- **Investment Costs Required to Initiate and Support Process**
  - Process support and coordination costs
  - Technical and documentation effort
  - Information dissemination costs

SEEDS First Public Workshop

SEEDS Formulation

Evaluator Information

Name:
Organization:
Current Activity:
Related Experience:
Job Focus: Choose one…
Email:

Software Reuse

Criteria \ Option	Status Quo Reuse	Improved Clone & Own	Open Source	Service Encapsulation	Product Lines
1. System cost savings	- / 0 / +	- / 0 / +	- / 0 / +	- / 0 / +	- / 0 / +
2. Flexibility & responsiveness	- / 0 / +	- / 0 / +	- / 0 / +	- / 0 / +	- / 0 / +
3. Increased effective & accountable community participation	- / 0 / +	- / 0 / +	- / 0 / +	- / 0 / +	- / 0 / +
4a. Suitability for ESE Mission Environment	- / 0 / +	- / 0 / +	- / 0 / +	- / 0 / +	- / 0 / +
4b. Suitability for ESE Science/Applications Environment	- / 0 / +	- / 0 / +	- / 0 / +	- / 0 / +	- / 0 / +
5. Investment cost	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H

Reference Architecture (Specificity)

Criteria \ Option	Status Quo Architecture	Notional	Concrete	Specific
1. System cost savings	- / 0 / +	- / 0 / +	- / 0 / +	- / 0 / +
2. Flexibility & responsiveness	- / 0 / +	- / 0 / +	- / 0 / +	- / 0 / +
3. Increased effective & accountable community participation	- / 0 / +	- / 0 / +	- / 0 / +	- / 0 / +
4a. Suitability for ESE Mission Environment	- / 0 / +	- / 0 / +	- / 0 / +	- / 0 / +
4b. Suitability for ESE Science/Applications Environment	- / 0 / +	- / 0 / +	- / 0 / +	- / 0 / +
5. Investment cost	L/M/H	L/M/H	L/M/H	L/M/H

Reference Architecture (Granularity)

Criteria \ Option	Coarse	Medium	Fine
1. System cost savings	- / 0 / +	- / 0 / +	- / 0 / +
2. Flexibility & responsiveness	- / 0 / +	- / 0 / +	- / 0 / +
3. Increased effective & accountable community participation	- / 0 / +	- / 0 / +	- / 0 / +
4a. Suitability for ESE Mission Environment	- / 0 / +	- / 0 / +	- / 0 / +
4b. Suitability for ESE Science/Applications Environment	- / 0 / +	- / 0 / +	- / 0 / +
5. Investment cost	L/M/H	L/M/H	L/M/H

Participation

SEEDS Formulation

- Positive Engagement of Responders

  "By the way, I thought the survey was well made and really made me think about the structure and content of provided interfaces/toolkits.Whoever is putting this together is asking the right questions"

- Good representation of DAACS, SIPS, ESIP-2s, ESIP-3s - Total of 18 responders

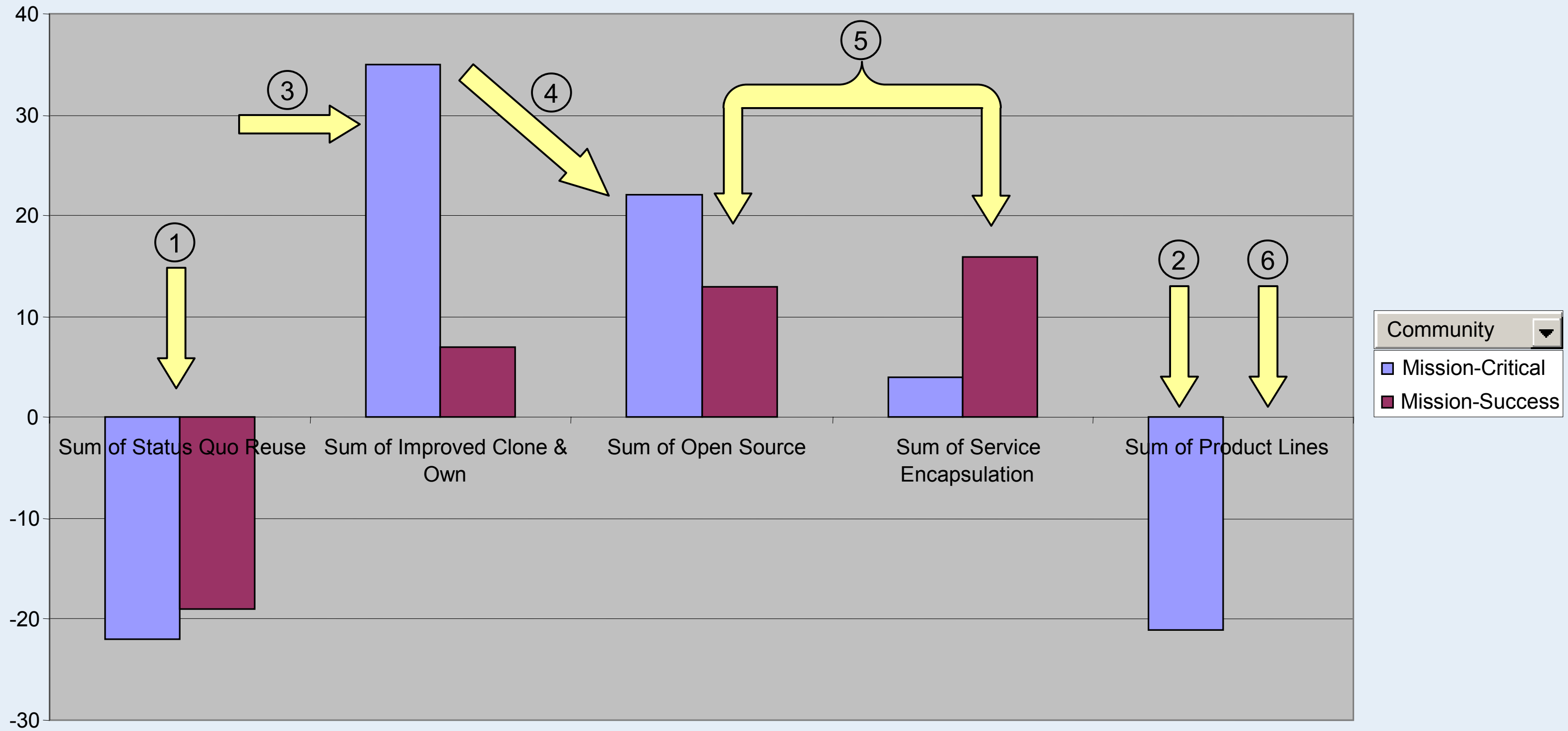
- To avoid one-size-fits-all analyzed community from differing viewpoints, strongest opinion differences fall along these lines:
  - mission-critical: driven by launch schedules and a need for daily, highly reliable production or archiving needs (e.g. SIPS, DAACs for standard products and high volume distribution)
  - mission-success: driven more by need for research in science, applications, or information systems, need to experiment with differing products, approaches, mechanisms and adapt to new understandings (e.g. ESIP-2s, -3s, analysis, etc.)

- Survey results will assume two approaches will be recommended, with each community providing guidance in their own areas
  - Community members "assigned" to groups by identification of "primary" funding source goals
  - Some community members participate strongly in both types of activities, for the purposes of this workshop, pick a "hat" to represent

Results to Date: Aggregate Community Opinion

- **General results**
  - The community agrees that the *Status Quo* is not satisfactory and that something needs to be done
  - The Community opinions regarding Reference Architecture alternatives were not as strong as they were regarding Reuse alternatives
  - There is a clear divergence of community-desired approaches, leading to the need for different processes for the two identified environments
- **Next slides provide**
  - Aggregate community opinion about identified options
    - Mission-critical community opinion
    - Mission-success community opinion
  - Aggregate community opinion about ***suitability*** of identified options
    - Self for self: Opinion of each community on the suitability of the options for their own environment
    - Cross opinion: Opinion of each community on the suitability of the options for the other environment

Aggregate Community Opinion about Reuse



① The *Status Quo* is not satisfactory for both communities.

Mission-Critical Community Opinion

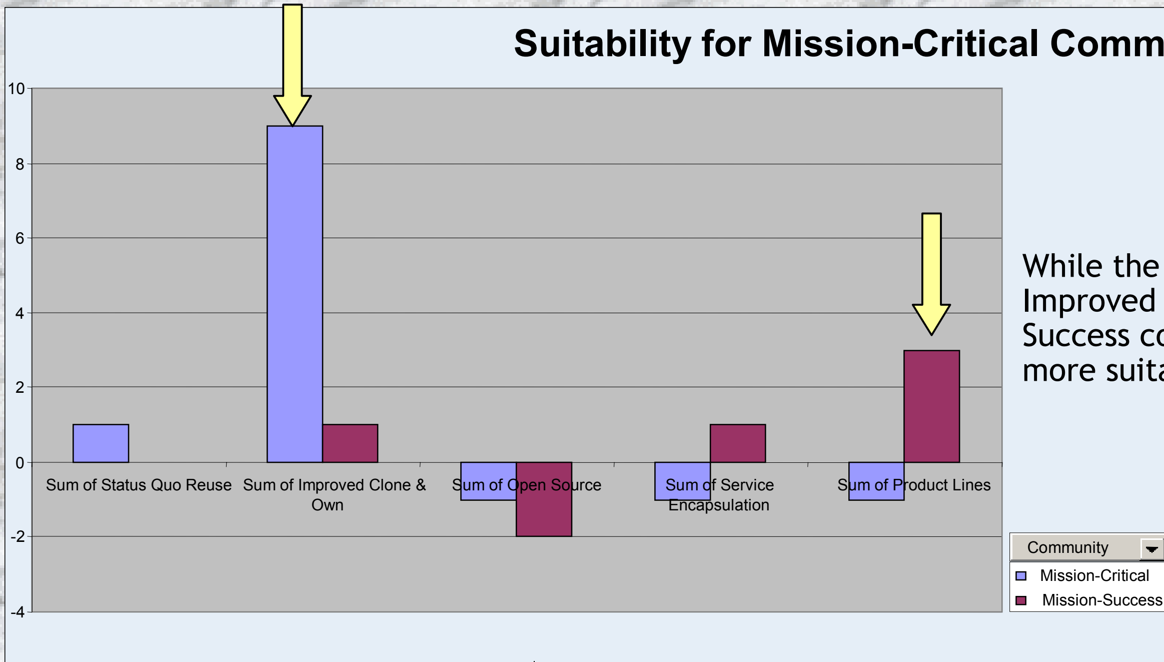
- ② Strong rejection of the Product Lines option because of association with past centralized development efforts.
- ③ Strong support for the Improved Clone & Own option.
- ④ Less support for Open Source option because of concerns about lack of control.

Mission-Success Community Opinion

- ⑤ Favoring of Service Encapsulation and Open Source options.
- ⑥ Disagreement about suitability of Product Lines.

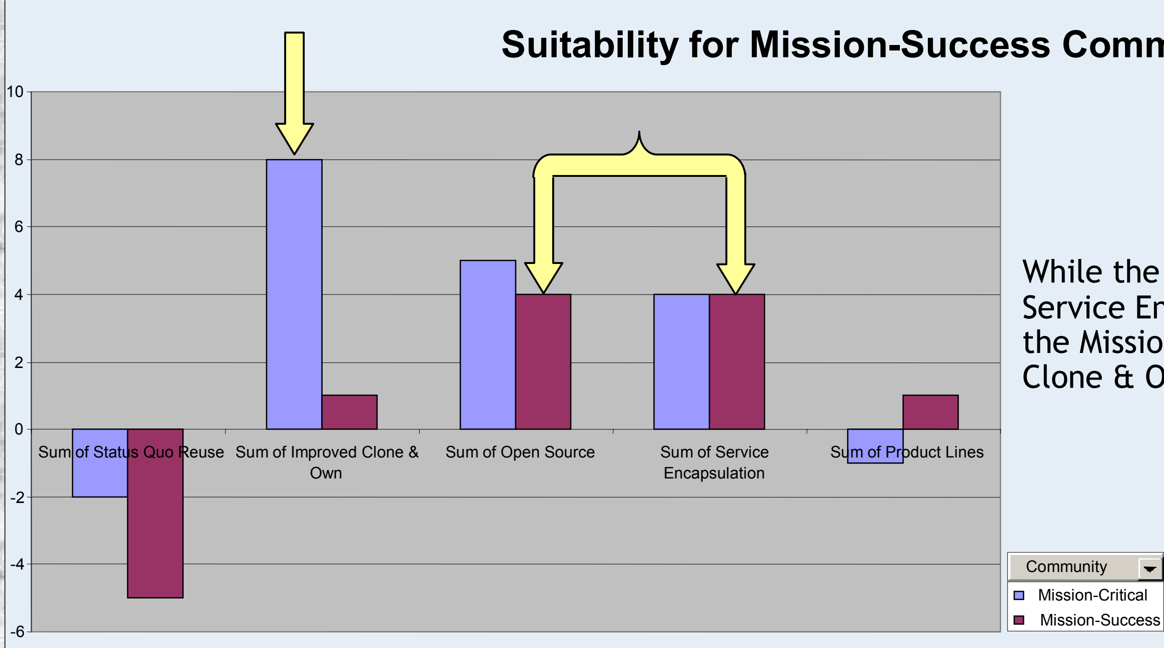
Self for Self and Cross Community Opinion about Reuse

SEEDS Formulation



While the Mission-Critical community strongly favors the Improved Clone & Own option for itself, the Mission-Success community considers the Product Lines approach more suitable for that environment.

The options preferred by each community differ from the one(s) proposed to it by outside communities.



While the Mission-Success community equally favors the Service Encapsulation and Open Source options for itself, the Mission-Critical community considers the Improved Clone & Own option more suitable for that environment.

Aggregate Community Opinion about Reference Architecture

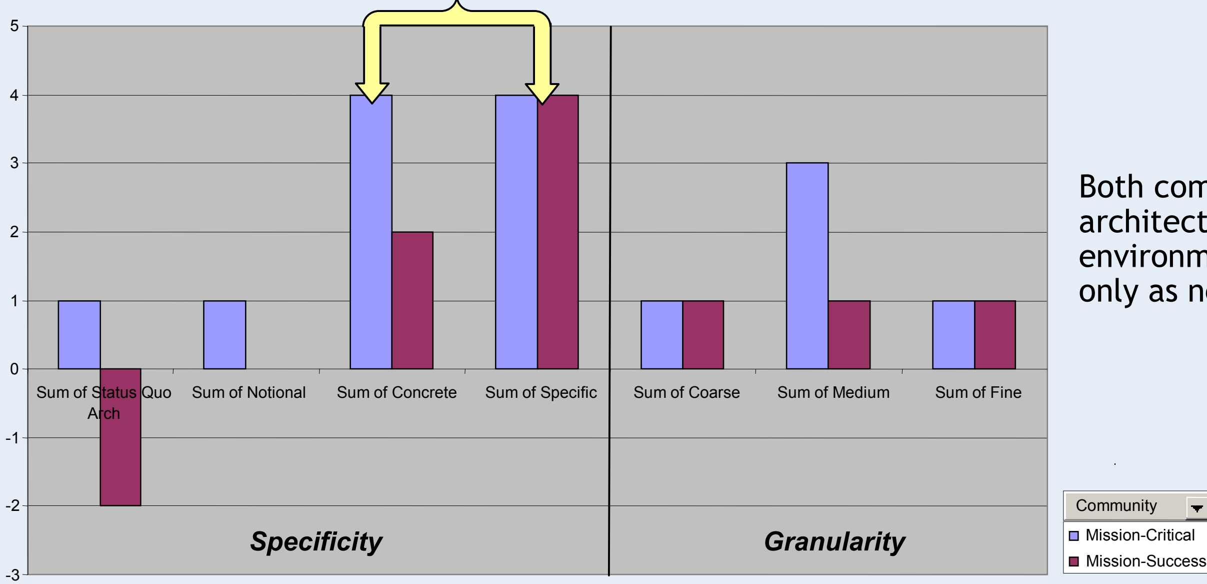
① The *Status Quo* considered not satisfactory, especially by the Mission-Success community.

② Support for a Notional or Concrete architecture which would drill down to more detail in selected functional areas.

③ Strong rejection of Fine grained architecture, emphasizing the community's interest in keeping the architecture at a high level of detail.

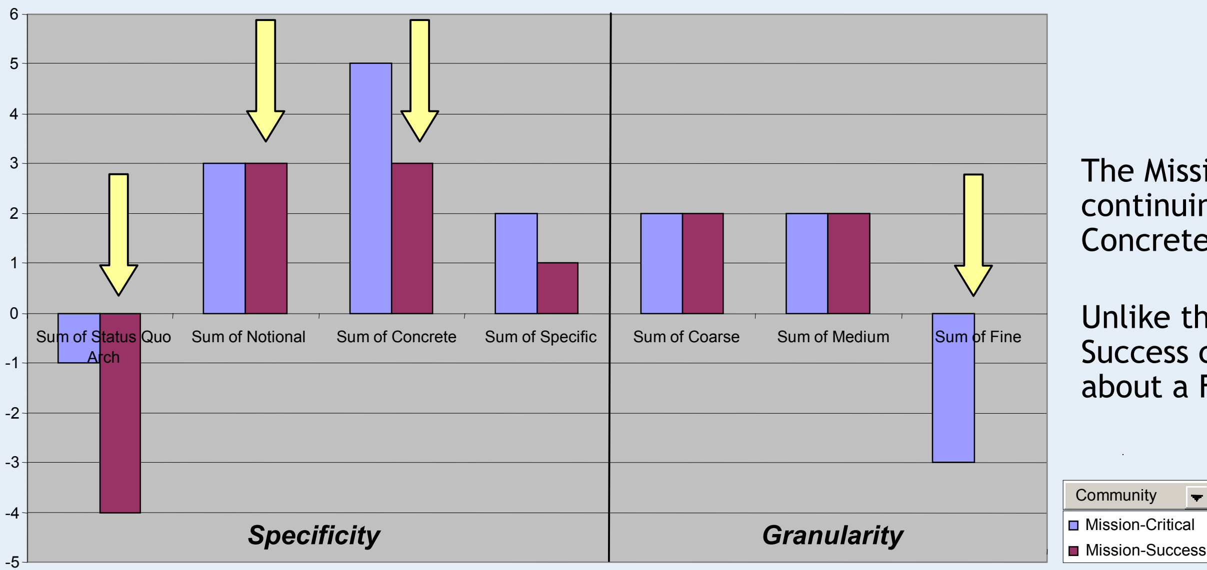
Self for Self and Cross Community Opinion about Reference Architecture

Suitability for Mission-Critical Community



Both communities find Concrete and Specific architectures suitable for the Mission-Critical environment, provided additional detail is added only as needed in selected functional areas.

Suitability for Mission-Success Community



The Mission-Success community is against continuing with Status Quo, favoring a Notional or Concrete architecture for itself.

Unlike the Mission-Critical community, the Mission-Success community seems to be in disagreement about a Fine architecture for itself.

Results to Date: Cost Sensitivity Analysis

- Purpose of Cost Sensitivity Analysis
  - Identify parameters that influence potential cost savings
  - Confirm cost savings opportunities for ESE
- Model
  - Accounts for the additional cost of developing reusable assets or making existing assets reusable (creating more generic designs, providing additional documentation, etc)
  - Accounts for the costs of reusing reusable assets (locating assets, evaluating assets, and integrating them into application, etc)
  - Accounts for the fact that a fixed percentage of each system is unique to that system
- Results
  - Significant cost savings can be achieved by increasing the percentage of development efforts employing reuse, and by increasing the amount of reuse within each development effort
  - By gradually increasing the reuse level over eight missions and by ensuring that all missions employ reuse, the ESE can free up a significant percentage of its custom development costs for other uses

Summary

SEEDS Formulation

- Dissatisfaction with *Status Quo* is clear
- Community Views about Reuse
  - Mission-Critical community strongly favors Improved Clone & Own
  - Mission-Success community views Open Source and Service Encapsulation with equal favor
- Community Views about Reference Architecture
  - Opinions not as strong as those about Reuse
  - Keep it coarse grained, notional with concrete details only in a limited set of functional areas
- Processes
  - Reuse does not happen by itself
  - One size does not fit all
  - Significant savings can be achieved by increasing reuse levels and mission participation rate
  - Use Reference Architecture to enable Reuse

Workshop Plan

- Get community input on guiding principles for setting up needed processes to move forward for each community

- My thoughts on what we are looking for (consistent with NEWDISS Pre-formulation document)
  - Interest in consensus-based processes done by actual stakeholders
  - Assure not one-size-fits-all – probably two working groups
  - Process is on-going, evolutionary – no big bang allowed
  - Interest in evolutionary test-bedding to prevent "systems-engineering-gone-mad" syndrome
  - Interest in leveraging work already done by other organizations if appropriate

- Areas
  - Contributing factors (issues/barriers & opportunities)
  - Guiding principles
  - Reuse program strategies
  - Reuse enablement strategies

Your input is needed. Please join us at 1 pm for our breakout session in Room Torrey Pines East.

Appendix

Definitions

- **Reuse**
  - Taking a functionality used in (or provided by) one system or mission and employing it in another system or mission
  - This functional capability can be in the form of code, or it can be design "artifacts" (e.g. architectures, software designs, ICDs, test plans, etc)
  - Broad definition for this study encompasses any means that avoids rebuilding a capability
- **Reference Architecture**
  - A generic architecture for use in particular domain (e.g. Earth science)
  - Used as a reference when developing a specific architecture
  - Provides a common reference to promote component reuse, reduce integration costs and promote interoperability
  - Focus is on enabling application (domain-specific vs. infrastructure) software reuse and application system openness
  - Could be high level or detailed